Amendments To The Specification

Please replace paragraph [0001] with the following re-written paragraph:

[0001] The subject matter of this application is related to the subject matter of U.S. Patent Application Serial No. [[_____]] 10/715,709 entitled "Method and System for Mapping Tags to Classes Using Name Spaces" filed ~~of even date herewith~~ on November 18, 2003, which application is assigned or under obligation of assignment to the same entity as this application, and which application is incorporated herein by reference.

Please replace paragraph [0025] with the following re-written paragraph:

[0025] As illustratively shown in Fig. 1, the binary representation 108 may consist of at least two components, namely information records 112 and structure records 114. Those records as illustrated may themselves embed fields including [[for]] identifiers to indicate whether a given record is of information, structure or other type. In embodiments size fields may likewise indicate the overall size of one or more records in the set of binary records 130, but formats may vary. In embodiments, the binary representation 108 may be generated from a live object tree as opposed to a stored file containing source XAML 102 or other data. The binary representation 108 may be transmitted to a binary reader 116 or receive data from a binary writer 118, for purposes of interfacing to other resources. Mapping engine 106 may likewise communicate with or make use of binary reader 116 or binary writer 118 to map or generate binary objects or outputs.

Please replace paragraph [0027] with the following re-written paragraph:

[0027] It may be noted that in embodiments, the binary representation 108 may be constructed such that the information records 112 are always available before they are referenced by any structure records 114. According to the invention in one regard, as noted the information records 112 may contain an ID field which is unique to a given binary representation 108 and may be used to uniquely identify assemblies, types and attributes. When structure records 114 are written out, they may contain an index of associated ID records for the type and attribute information. For example, consider the following XAML expression, which may form an instance of source XAML 102:

Expression 1

```
<DockPanel  xmlns=" http://shemeasschemas.microsoft.com/2003/xaml">
    <Button Background="Blue"/>
</DockPanel>
```

Please replace paragraph [0028] with the following re-written paragraph:

[0028] Upon receipt by the mapping engine 106, this source XAML 102 may be used to generate the associated set of binary records 130 illustrated in Fig. 4. In the illustrative record, it may be noted that every information record 112 is written out before it is referred to. When reading the set of binary records 130 associated with binary representation 108 via binary reader 116 or otherwise, ~~hoisted type,~~ assembly and attribute information may be used to build tables of information used to construct a representation of the object tree 110 and set properties.

Please replace paragraph [0033] with the following re-written paragraph:

[0033] If a type implements serialization interfaces configured to be consistent with the invention, it may be called upon when a stream of binary representation 108 is

constructed to serialize itself directly into the stream in its own custom defined format resulting in compact storage and transmission requirements. The Length (or other dimensional) type can exploit this type of capability effectively, resulting in typical cases in a requirement of only 1 byte to specify the length of an object in a binary stream. Fig. 5 for example shows illustrative code for performing a serialization for a Length type. Other implementing code is possible. The binary representation 108 and associated [[may]] records may thus likewise be optimized by generating a type index for novel types (such as Length) upon encountering the first instance of those types. Loading times may also therefore be enhanced.

Please replace the Abstract with the following re-written Abstract:

ABSTRACT OF THE DISCLOSURE

A system and related techniques accept extensible application markup language (XAML) inputs corresponding to object trees, such as those representing user interface elements, and map those inputs to a binary construct. A mapping engine may generate tokenized binary outputs representing the XAML file input and ultimately the associated user interface or other object. The binary representation generated by the mapping engine may be optimized in multiple ways, including to encode dimension information such as length, width etc. of dialog boxes or other elements in the binary representation without a necessity for explicit definition. Other optimizations include the type indexing of data types when a novel instance of the type is first encountered, and embedding loader definitions to load the object or data reflected in the binary representation without having to do a lookup against loader lists.